

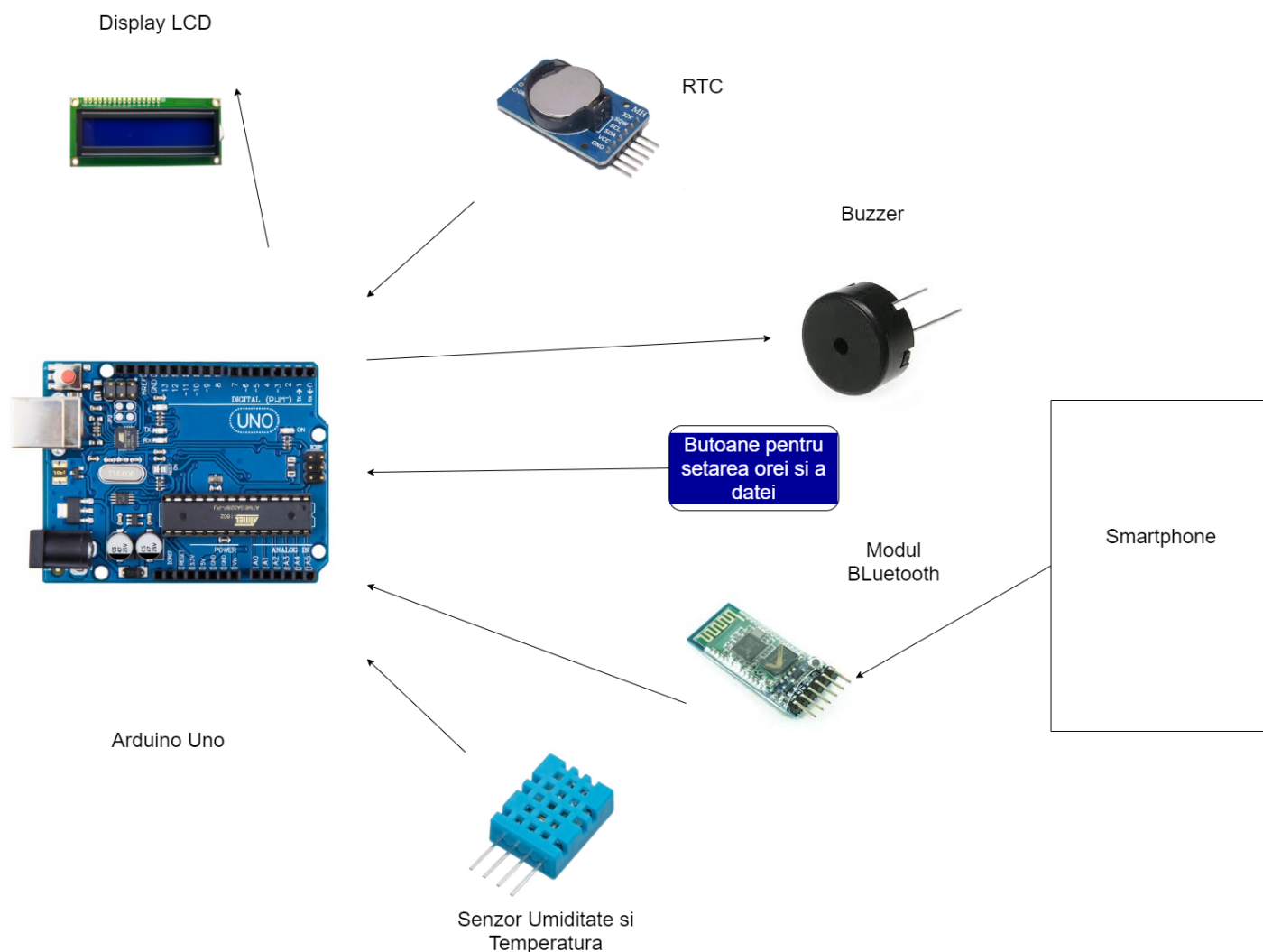
Ceas digital cu alarma

Autor: Descultu Cristian Petrisor

Introducere

Scopul proiectului este de a realiza un ceas digital cu alarma si care iti poate oferi si temperatura si umiditatea din aer. De asemenea, alarma acestuia va putea fi programata si controlata prin telefon, prin intermediul unei aplicatii. Pe langa acestea s-ar putea introduce si un senzor de luminozitate astfel incat atunci cand se va trece cu mana peste acesta, alarma s-ar putea inchide.

Descriere generala



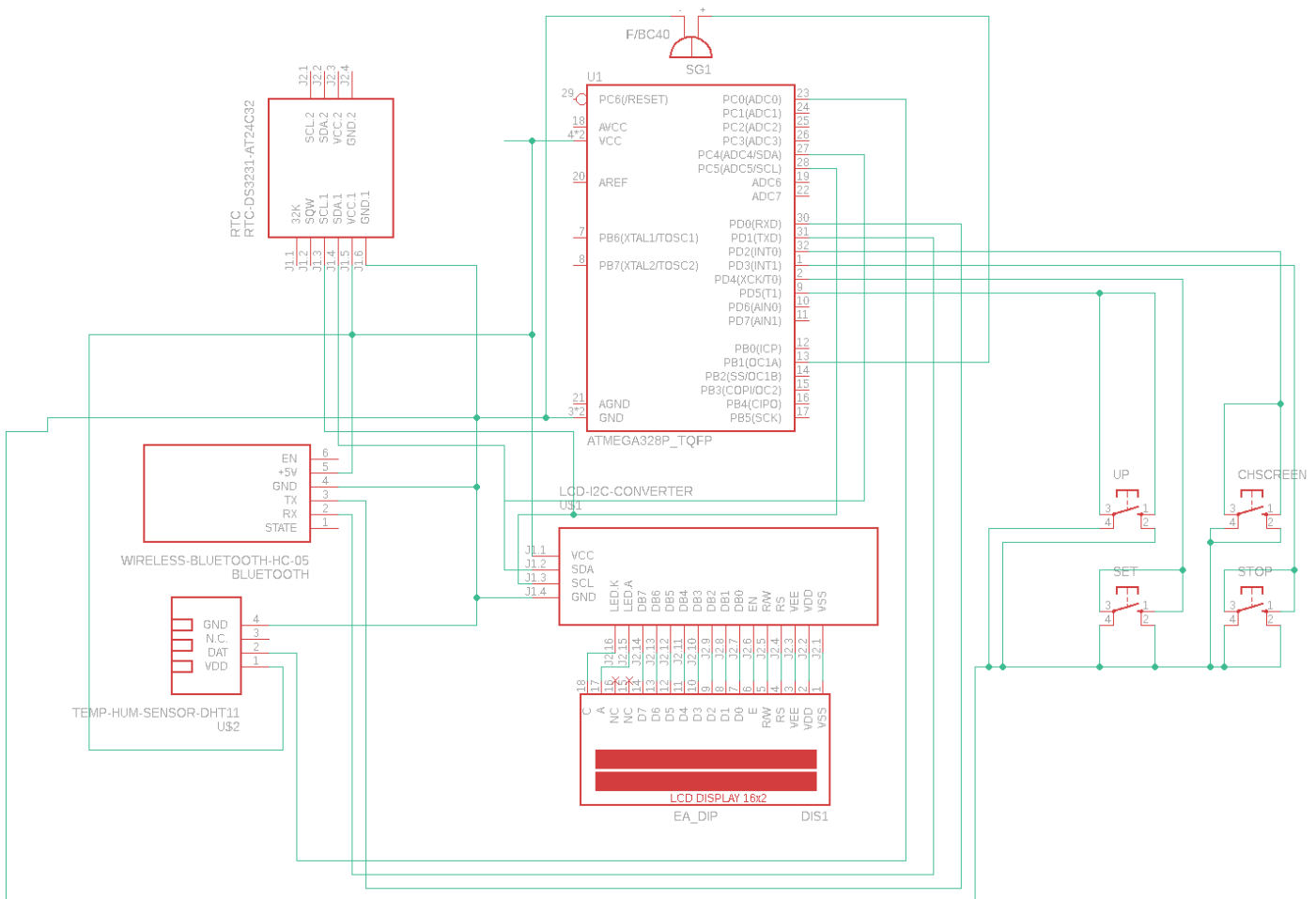
Microcontroller-ul va trimite date catre LCD pentru a putea fi afisate. Datele pentru temperatura si umiditate vor fi preluate de la senzorul DHT11. RTC are scopul de nu pierde datele legate de timp si data chiar daca microcontroller-ul nu este conectat la o sursa de curent. Butoanele vor modifica valorile transmise catre LCD, avand scop de setarea a orei si a datei. Modulul bluetooth va fi folosit pentru a asigura conexiunea cu smartphone-ul pentru ca user-ul sa-si poata seta alarma. Buzzer-ul este folosit pe post de alarma, astfel incat atunci cand anumite valori de timp ajung la ora si data setata de alarma, buzzer-ul va porni si va emite efecte sonore timp de cateva secunde sau pana cand utilizatorul va acoperi usor senzorul de lumina, in cazul in care acesta va fi folosit.

Hardware design

Lista Componente

1. Breadboard
2. fire de legatura
3. RTC
4. Display LCD 16x2
5. Modul bluetooth hc-05
6. butoane
7. senzor de temperatura si umiditate DTH11
8. buzzer

Schema Electrica

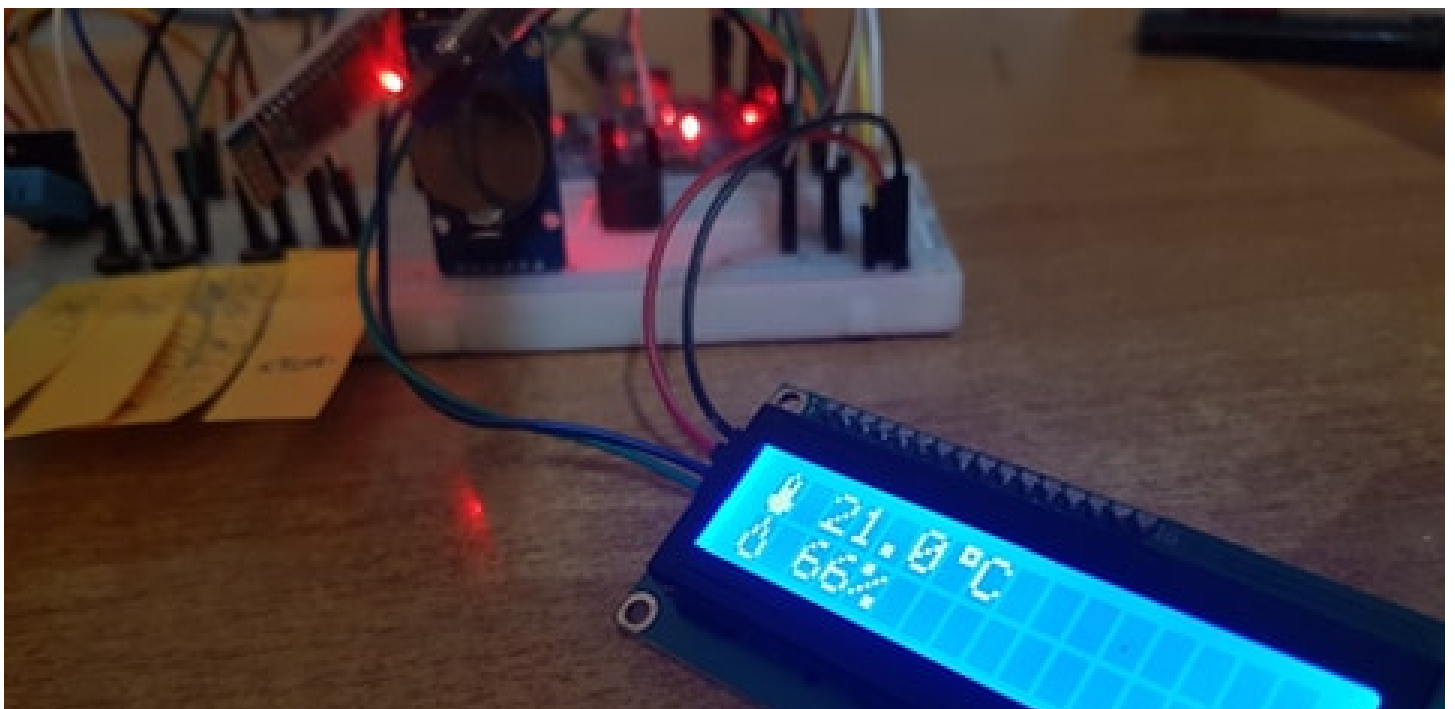


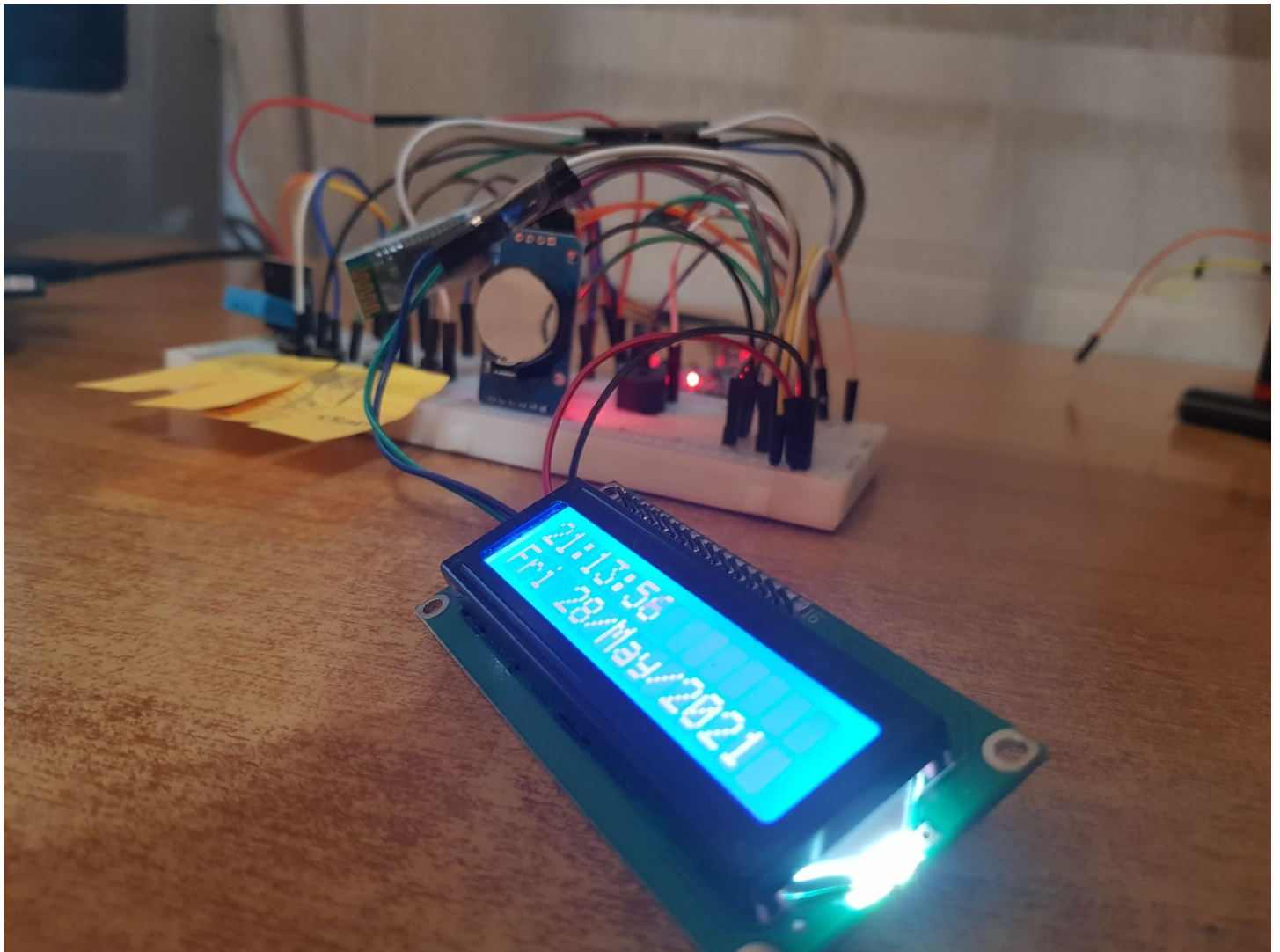
Software Design

- Pentru dezvoltarea aplicatiei am folosit Arduino IDE si MIT App Inventor pentru realizarea unei aplicatii android care sa poata comunica cu Arduino.
- Bibliotecile folosite sunt "LiquidCrystal_I2C.h", "dht.h", "RTCLib.h", "Wire.h", "stdio.h", "stdlib.h".
- Am folosit o structura de date pentru a retine datele legate de alarma si de statusul acesteia.

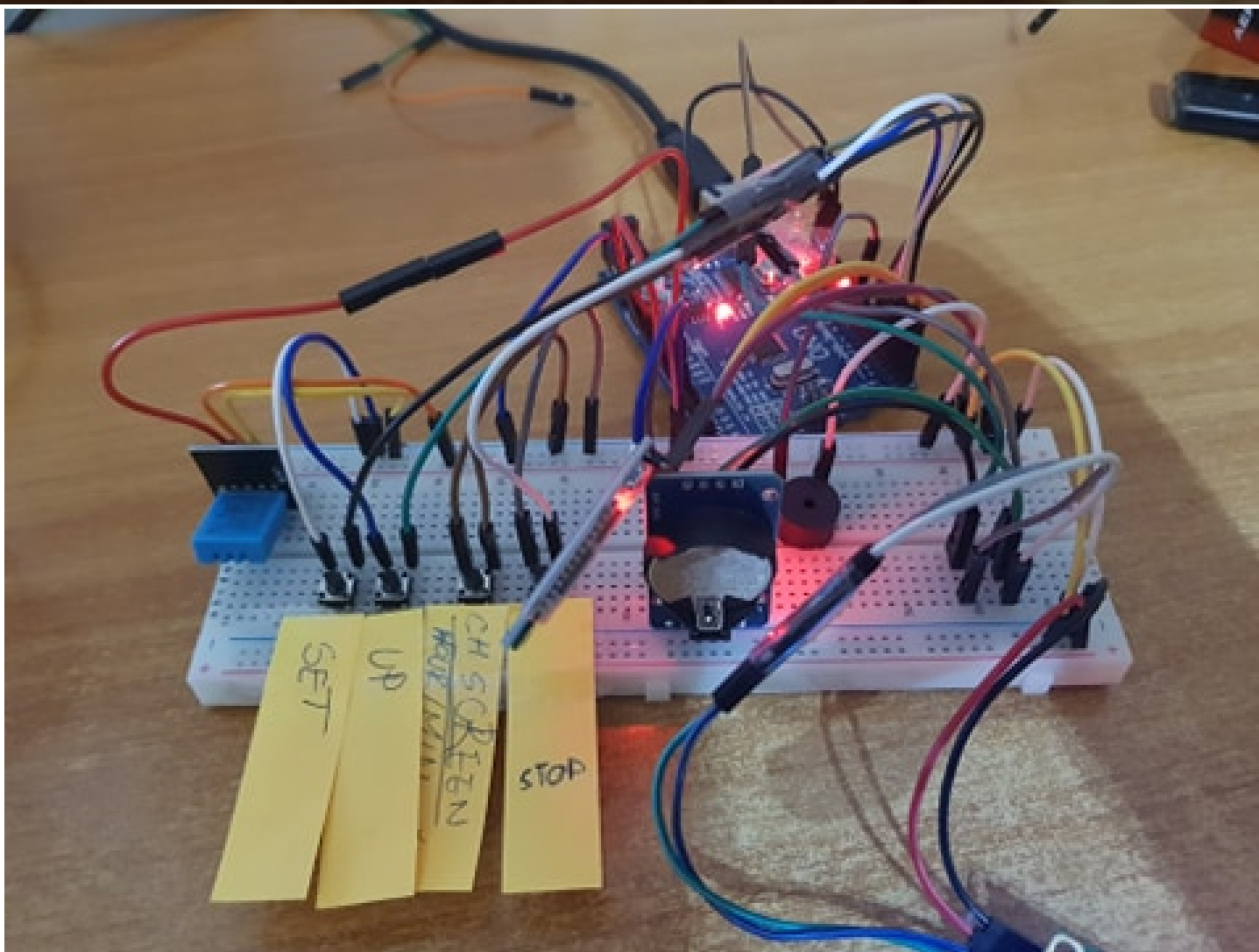
Functionare Ceas

- Se porneste interfata seriala si LCD-ul, iar pinii de in/out sunt setati
- RTC-ul este pornit (daca nu este sincronizat in concordanta cu ora sistemului, acest lucru se va face decommentand linia marcata
- Se verifica ce tip de butoane au fost apasate pentru a intoarce comportamentul dorit
- Alarma poate fi setata fie apasand butonul "SET", urmat de "UP" pentru setarea orei/minutului, dupa care se va apasa "CHScreen", pentru a trece la urmatorul element sau pentru a finaliza setarea, fie prin intermediul aplicatiei android.
- Pentru oprirea alarmei se va apasa butonul "STOP".
- Se verifica in mod constant daca exista date pe interfata seriala (s-au trimis date de pe smartphone prin bluetooth catre arduino).









Rezultate Obținute

S-a obtinut un ceas digital care afiseaza pe ecranul principal detalii despre data si ora, acesta putand fi schimbat la apasarea unui buton. Acesta are incorporat si o alarma care se poate seta fie fizic, apasand butoanele de pe arduino, fie remote prin intermediul unei aplicatii android. Cand se ajunge la ora si minutul setate pentru alarma, se va auzi "The Imperial March". Alarma se poate opri la apasarea butonului de pe breadboard.

Demo [<https://youtu.be/51THLYmIkQc>]

Concluzii

Dispozitivul functioneaza impecabil, singura modificare pe care as adauga-o pe viitor ar fi un "blinking digits" in momentul in care se seteaza ora/minutul de catre user, pentru a fi mai intuitiv la care categorie se afla.

Download

alarm_clock.zip - Arhiva contine sursele, README, bibliotecile folosite si aplicatia android.

Jurnal

Am inceput prin a ma obisnui cu fiecare componenta in parte, astfel ca am conectat separat Display-ul, RTC-ul si senzorul DHT11 si m-am jucat cu ele cu scopul de a ma familiariza si de a-mi pune niste baze. Ulterior, dupa ce am considerat ca sunt pregatit, am inceput sa iau proiectul pe bucatele. Am conectat si configurat LCD-ul, doar afisand un mesaj. Apoi am adaugat senzorul la proiect, am citit si prelucrat datele si am verificat daca se trimit bine la LCD. Cand am fost multumit si am facut mici modificari la cum sa afisez informatiile, am adaugat si RTC-ul. In urma informatiilor gasite, acesta trebuie initializat o singura data, de aceea in cod am comentat linia respectiva, pentru a fi in continuare acolo in cazul in care crapa ceva(am patit:)). Am testat ca totul merge ok(opream si porneam alimentarea sa vad ca datele temporale sunt retinute in RTC). Nefiind multumit de modul in care se afisa si de faptul ca avand foarte multe delay-uri in cod, era greu de apasat un buton pentru oprirea alarmei, am ajuns de la un display care se schimba automat, la unul controlat din butoane. Am adaugat 4 butoane, pentru schimbarea ecranului, setarea alarmei si oprirea acesteia. Cand am rulat din nou cateva teste fie setand alarma din cod, fie din butoane, mi-am dat seama ca bipaitul acela monoton al buzzer-ului nu-i foarte incantator asa ca am decis sa iau o configuratie de pe net pentru buzzer pentru a recrea o melodie. Ultima etapa a fost sa invat sa folosesc site-ul MIT APP INVENTOR (se poate gasi la resurse) pentru dezvoltarea unei aplicatii android minimaliste pentru a seta alarma.

Bibliografie/Resurse

1. Link proiect [<https://ocw.cs.pub.ro/courses/pm/prj2021/abasoc/ceasdigital>]

Resurse Hardware

1. Tutorial conectare LCD [<https://www.instructables.com/How-to-Connect-I2C-Lcd-Display-to-Arduino-Uno/>]
2. Tutorial conectare RTC [<https://create.arduino.cc/projecthub/electropeak/interfacing-ds1307-rtc-module-with-arduino-make-a-reminder-08cb61>]
3. Tutorial Bluetooth [<https://create.arduino.cc/projecthub/mayooghgirish/arduino-bluetooth-basic-tutorial-d8b737>]
4. Tutorial DHT11 [<https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239>]

Resurse Software

1. Tutorial conectare LCD [<https://www.instructables.com/How-to-Connect-I2C-Lcd-Display-to-Arduino-Uno/>]
2. Tutorial conectare RTC [<https://create.arduino.cc/projecthub/electropeak/interfacing-ds1307-rtc-module-with-arduino-make-a-reminder-08cb61>]
3. Tutorial Bluetooth [<https://create.arduino.cc/projecthub/mayooghgirish/arduino-bluetooth-basic-tutorial-d8b737>]
4. melodie alarma [<https://create.arduino.cc/projecthub/HiHiHiHiiHiiH/star-wars-on-a-buzzer-0814f2>]
5. dezvoltare aplicatie android [<https://appinventor.mit.edu/>]
6. tutorial app inventor [<https://www.youtube.com/watch?v=pZWdFEzCZP0>]
7. Tutorial DHT11 [<https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239>]

pm/prj2021/abasoc/ceasdigital.txt · Last modified: 2021/05/28 21:27 by cristian.descultu